In the Specification:

Please amend the specification with the following replacement paragraphs:

[0001] Many types of computer-related apparatus utilize storage devices to store data. For example, data may be stored temporarily in networking devices through which the data passes. Various constraints typically apply to such apparatus. For example, especially in the case of high-bandwidth or high-capacity apparatus, the bandwidth or capacity requirements of the apparatus may exceed the capabilities of available storage devices. To overcome the limitations of the storage device, multiple storage devices may be employed and used together to provide higher combined bandwidth and/or capacity. However, even when multiple storage devices are used together, inherent characteristics of the storage devices can affect the efficiency with which data is stored or retrieved. For example, storage devices typically store or retrieve data in quantities of a unit of access. As one particular example, some storage devices, such as some memory devices, provide a burst access mode in which bursts of some number of bytes of data may be transferred in a single memory operation.

[0002] In a broader sense, data structures comprising data may be stored in and retrieved from the storage devices. Such storage and retrieval may be performed in increments of the data of such data structures. Such increments, that may or may not be compatible with read and write access, may be provided to fixed-size portions of a data structure (e.g., frames) that are stored in units larger than the units of access. For example, storage devices based on dynamic random access memory (DRAM) devices have a characteristic of burst access, which defines the size of the unit of access and predefined starting points of the access (e.g., bursts of 16 bytes on predefined 16 byte boundaries). The bandwidth available from one device is often less than that specified by system requirements.

[0003] Figure 2 is a memory map diagram illustrating a method and apparatus for efficiently accessing data segments having arbitrary alignment with the memory structure in which they are stored in accordance with at least one embodiment of the present invention. The memory structure 201 is organized to have a memory structure width 202, which, in the illustrated example, is 16 bits. The memory structure 201 begins at a beginning memory location 203 and ends at an ending memory

location 204. A number of modes of memory access is provided, where that number of modes of memory access is at least the number of storage devices, which, in the illustrated example, is four (e.g., A, B,C, and D). For example, a first mode 205 of memory access spans memory units of access 212, 213, 214, and 215 of storage devices A, B, C, and D, respectively. A second mode 206 of memory access spans memory units of access 213, 214, 215, and 216 of storage devices B, C, D, and A, respectively. A third mode 207 of memory access spans memory units of access 214, 215, 216, and 217 of storage devices C, D, A, and B, respectively. A fourth mode 208 of memory access spans memory units of access 215, 216, 217, and 218, corresponding to storage devices D, A, B, and C, respectively. Memory unit of access 219 follows memory unit of access 218.

[0004] For the particular example given, a typical access is 48 bytes (e.g., the payload of an ATM cell). There is no alignment of 48 bytes that cannot be completely retrieved in a single read of a 64-byte burst access in accordance with at least one embodiment of the present invention. Consequently, additional overhead, in terms of extra read cycles required to read data that has been re-aligned with respect to access alignment (i.e., access starting point and access burst size) of the memory structure, may be minimized.

[0005] At least one embodiment of the present invention is advantageous in that it is able to utilize ~~similar~~ storage devices (e.g., A, B, C, and D) ~~as are~~ similar to those typically used, thereby allowing a conventional memory hardware architecture to be preserved. However, by utilizing a few additional address bits, at least one embodiment of the present invention may be used to maximize memory bandwidth efficiency regardless of alignment of portions of the memory structure being accessed with boundaries within the memory structure observed during memory accesses.

[0006] Optionally, the total memory bandwidth retrieved is ~~store~~ stored contiguously in memory. Optionally, the desired data is a contiguous block of data in memory.

[0007] Each storage device provides its own range of hardware memory locations, which may be mapped into system memory locations of a system. For example, D storage devices, each providing H memory locations, may be combined with appropriate memory mapping to yield a system having $S = D \times H$ system memory locations. In order to maximize total memory bandwidth, it is advantageous to map the hardware memory locations of multiple storage

devices such that, for access to a contiguous range of system memory locations greater than a number of storage device hardware memory locations of a single storage device that can be accessed during a single memory access cycle, the access spans multiple storage devices. ~~it~~ It is beneficial to map system memory locations such that portions of the ranges of hardware memory locations of the multiple storage devices appear in a sequential pattern. For example, it is desirable to map a portion of the hardware memory locations of a first storage device (e.g., A) having a size up to as much data as may be read from such a first device during one memory access operation followed by a portion of the hardware memory locations of a second storage device (e.g., B) having a size up to as much data as may be read from such a second device during one memory access operation followed by instances of portions of hardware memory locations of any other storage devices in sequence.

[0008] In accordance with such an example, addressing may be provided by expressing an address as a concatenation of a block pointer, an instance selector, and a bank selector. The block pointer 525 comprises ~~a~~ one or more bits, such that a value represented by the one or more bits points to the beginning of the block 501, which may be one of many such blocks within a data structure stored in the storage devices. The instance selector comprises one or more bits, such that a value represented by the one or more bits selects among instances of units of access, such as 520-523. In accordance with a preferred data structure organization, the instance selector can remain the same for accesses to both of banks 518 and 519. The bank selector comprises one or more bits, such that a value represented by the one or more bits selects among banks, such as banks 518 and 519. Since a plurality of storage devices is typically similarly configured, the bank selector may be used in a common manner for all such storage devices.